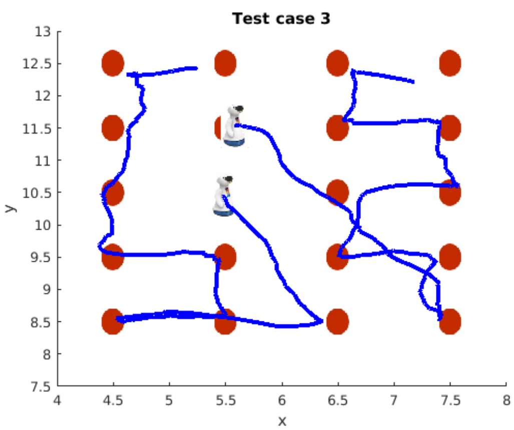
Market-Based Coordination for Social Robots in Human-Populated Environment

Stefano Savaré

Professor : Alcherio Martinoli
Assistant(s) : Zeynab Talebpour

In the last 30 years the research in the robotic field has continuously grown and different aspects of this problem have been studied. The increased computing power and the growing knowledge make possible to investigate multi robot-multi task configurations. It becomes necessary for this objective to have a good framework that allows the robot to take decision considering the status of the others. For this purpose I implemented a framework called "Hoplites", that is based on the idea of dynamically changing the



Standard test case studied

complexity of the interactions between the robots depending on the environment and robots situation. Alongside Hoplites, that deals only with the communication aspects, I implemented a simple planner, based on a one-step optimization algorithm. I implemented all of this using the Python language and ROS (Robot Operating System); I finally tested it both on the Webots simulator and on the MOnarCH robots, the robots for which the code has been created.

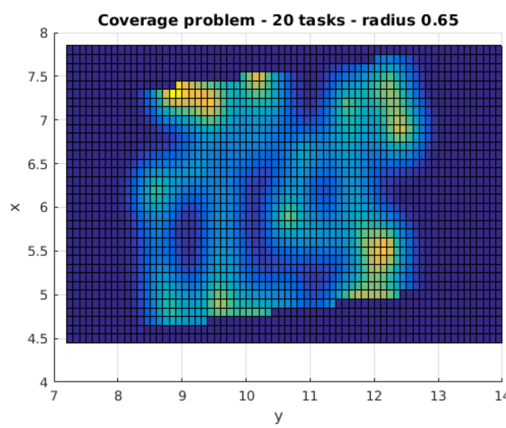
The standard test case scenario, that you can see in the first figure, consists in asking a set of robots to reach a set of tasks, in different positions, while minimizing some functionals. This test case has been developed in 2 main variants:

1°) Task allocation problem. It consists in making the robot to reach all the tasks, only one time, minimizing the overall time.

2°) Persistent coverage problem. It consists in covering an area in the best way possible.

The results that I got depend on the problem considered.

The robots behave like expected in the task allocation problem. There are no performance differences between the simulator and the real robots and the time to complete the task is minimized, compatibly with the planner used.



Visual results of the coverage problem

In the persistent coverage problem the performances are less optimal, since there is a lot of variance in the coverage level across the environment, as you can see in the last figure. However the solution that I got is, in any case, acceptable for a lot of possible applications. Possible improvements have to be found in the part of the algorithm that deals with the coverage problem and not in the Hoplites framework, that behaves well.

Possible improvements to my implementation could be a more advanced planner and the addition of sources of conflicts, like human presence, that I did not consider.